Color Space Converter

MegaCore Function User Guide

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com

MegaCore Version: 2.2.0
Document Version: 2.2.0
Document Date: June 2004

Copyright © 2004 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

I.S. EN ISO 9001

Contents

About this User Guide v
- Revision History v
- How to Contact Altera v
- Typographic Conventions vi

Chapter 1. About this MegaCore Function
- Release Information 1–1
- Device Family Support 1–1
- Introduction 1–2
- New in Version 2.2.0 1–2
- Features 1–2
- General Description 1–3
  - OpenCore Plus Evaluation 1–3
  - DSP Builder Support 1–3
- Performance 1–4

Chapter 2. Getting Started
- System Requirements 2–1
- Design Flow 2–1
- Obtain & Install the CSC MegaCore Function 2–2
  - Download the CSC MegaCore Function 2–2
  - Install the CSC MegaCore Function Files 2–3
  - Directory Structure 2–4
- CSC MegaCore Function Walkthrough 2–4
  - Create a New Quartus II Project 2–5
  - Launch IP Toolbench 2–6
  - Step 1: Parameterize 2–8
  - Step 2: Set Up Simulation 2–12
  - Step 3: Generate 2–14
- Simulate the Design 2–16
- Compile the Design 2–16
- Program a Device 2–17
- Set Up Licensing 2–17
  - Append the License to Your license.dat File 2–17
  - Specify the License File in the Quartus II Software 2–18

Chapter 3. Specifications

Functional Description 3–1
Signals 3–2
OpenCore Plus Time-Out Behavior 3–2
Parameters 3–3
Signals 3–5
MegaCore Verification 3–5
References 3–6

About this User Guide

Revision History

The table below displays the revision history for the chapters in this User Guide.

Chapter	Date	Version	Changes Made
All	June 2004	2.2.0	Updated the User Guide for version 2.2.0 of the Color Space Converter (CSC) MegaCore function. Edited for standards conformance and included all new screenshots.
All	April 2004	2.1.0	New document for product version 2.1.0. Added new CSC MegaCore function features plus support for Altera® Stratix®, Stratix II and Cyclone™ devices, IP functional simulation models, and the OpenCore Plus® evaluation feature. Reorganized content to new chapter scheme.

How to Contact Altera

For technical support or other information about Altera products, go to the Altera world-wide web site at www.altera.com. You can also contact Altera through your local sales representative or any of the sources listed below.

Information Type	USA & Canada	All Other Locations
Technical support	www.altera.com/mysupport/	www.altera.com/mysupport/
	800-800-EPLD (3753) 7:00 a.m. to 5:00 p.m. Pacific Time	+1 408-544-8767 7:00 a.m. to 5:00 p.m. (GMT -8:00) Pacific Time
Product literature	www.altera.com	www.altera.com
Altera literature services	lit_req@altera.com	lit_req@altera.com
Non-technical customer service	800-767-3753	+ 1 408-544-7000 7:00 a.m. to 5:00 p.m. (GMT -8:00) Pacific Time
FTP site	ftp.altera.com	ftp.altera.com

Typographic Conventions

This document uses the typographic conventions shown below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.*
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword SUBDESIGN), as well as logic function names (e.g., TRI) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
	The hand points to information that requires special attention.
	The caution indicates required information that needs special consideration and understanding and should be read prior to starting or continuing with the procedure or process.
	The warning indicates information that should be read prior to starting or continuing the procedure or processes
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.

Chapter 1. About this MegaCore Function

Release Information

Table 1–1 provides information about this release of the Color Space Converter (CSC) MegaCore® function.

Table 1–1. CSC MegaCore Function Release Information

Item	Description
Version	2.2.0
Release Date	June 2004
Ordering Code	IP-CSC
Product ID(s)	0x03
Vendor ID(s)	6AF7

Device Family Support

MegaCore functions provide either full or preliminary support for target Altera device families, as described below:

- *Full support* means the MegaCore function meets all functional and timing requirements for the device family and may be used in production designs
- *Preliminary support* means the MegaCore function meets all functional requirements, but may still be undergoing timing analysis for the device family; it may be used in production designs with caution.

Table 1–2 shows the level of support offered by the CSC MegaCore function to each Altera device family.

Table 1–2. Device Family Support

Device Family	Support
Stratix™ II	Full
Stratix GX	Full
Stratix	Full
Cyclone™	Full
Cyclone II	Full

Table 1–2. Device Family Support

Device Family	Support
Mercury™	Full
ACEX® 1K	Full
APEX™ II	Full
APEX 20KE & APEX 20KC	Full
APEX 20K	Full
FLEX® 10K	Full
Other device families	No support

Introduction

A color space is a method for precisely specifying the display of color using a three-dimensional coordinate system. Different color spaces are best for different devices, such as RGB (red-green-blue) for CRT monitors or YCbCr (luminance-chrominance) for digital television. The CSC MegaCore function provides a flexible and efficient means to convert image data from one color space to another, and is suitable for use in a wide variety of image processing and display applications.

New in Version 2.2.0

- Support for Cyclone II devices
- DSP Builder support
- IP Toolbench dialog box updates

Features

- Computes one output per clock cycle
- Typically runs at clock speeds over 200 MHz in Stratix devices
- Supports a variety of conversion functions:
  - Studio video R'G'B' to Y'CbCr
  - Y'CbCr to studio video R'G'B'
  - Computer R'G'B' to Y'CbCr
  - Y'CbCr to computer R'G'B'
  - Y'IQ to Y'UV
  - Computer R'G'B' to Y'UV
  - Y'UV to computer R'G'B'
  - User-specified conversion constants
- Supports signed and unsigned input data widths from 2 to 32 bits
- Provides user-selectable output precision via parameterizable rounding, saturation, and truncation

General Description

Color space conversion is often necessary when transferring data between devices that use different color space models. For example, to transfer a television image to a computer monitor, you may need to convert the image from the Y'CrCb color space to the R'G'B' color space. Conversely, transferring an image from a computer display to a television set may require a transformation from the R'G'B' color space to the Y'CrCb color space. You can use the CSC MegaCore function to perform these types of color transformations for a variety of applications, including image filtering, machine vision, and digital video.

OpenCore Plus Evaluation

With the Altera free OpenCore Plus evaluation feature, you can perform the following actions:

- Simulate the behavior of a MegaCore function within your system
- Verify the functionality of your design, as well as quickly and easily evaluate its size and speed
- Generate time-limited device programming files for designs that include MegaCore functions
- Program a device and verify your design in hardware

You only need to purchase a license for the MegaCore function when you are completely satisfied with its functionality and performance, and want to take your design to production.

For more information on OpenCore Plus hardware evaluation using the CSC MegaCore function, see "OpenCore Plus Time-Out Behavior" on page 3–2 and application note *AN 320: OpenCore Plus Evaluation of Megafunctions* from the Altera web site.

DSP Builder Support

Altera's DSP Builder shortens DSP design cycles by helping you create the hardware representation of a DSP design in an algorithm-friendly development environment.

You can combine existing MATLAB/Simulink blocks with Altera DSP Builder/MegaCore blocks to verify system level specifications and generate hardware implementations. After installing this MegaCore function, a Simulink symbol of this MegaCore function appears in the Simulink library browser in the MegaCore library from the Altera DSP

Builder blockset. To use this MegaCore function with DSP Builder, you require DSP Builder v2.2.0 or higher and the Quartus II sofware version 4.1 or higher.

For more information on DSP Builder, refer to the *DSP Builder User Guide* and the *DSP Builder Reference Manual*.

Performance

The CSC MegaCore function yields efficient implementation results, and its parameterization allows you to fine tune these results to achieve the utilization and performance you require.

Table 1–3 shows the resource utilization and maximum clock frequency for several sample implementations in different device families. These all use the computer R'G'B' to Y'CrCb color model conversion function, and differ in their choice of hardware implementation method, input data width, and pipeline level. The figures were generated using the Quartus® II software version 4.1.

Table 1–3. CSC MegaCore Function Performance Samples

Family	Speed Grade	Hardware Implementation	Width of Input Data	Pipeline Level	LEs	18*18 Mults	f_{MAX} (MHz)	Tpd (ns)
Stratix	-5	Distributed Arithmetic in LUTs	8	6	297	0	314	–
		Multipliers using LUTs	8	6	287	0	271	–
		Multipliers using DSP blocks	8	6	204	9	260	–
		Multipliers using LUTs *(2)*	8	0	103	0	180	5.5
Cyclone II	-3	Distributed Arithmetic in LUTs	8	6	292	0	216	–
		Multipliers using LUTs	8	6	299	0	308	–
		Multipliers using DSP blocks	8	6	243	9	230	–
		Multipliers using LUTs *(2)*	8	0	78	0	175	5.7
Stratix II *(1)*	-3	Distributed Arithmetic in LUTs	10	6	520	0	380	–
		Multipliers using LUTs	10	6	490	0	410	–
		Multipliers using DSP blocks	10	6	150	9	270	–
		Multipliers using LUTs *(2)*	10	0	87	0	240	4.1

Notes to Table 1–3:

(1) The Quartus II software reports the number of adaptive look-up tables (ALUTs) that the design uses in Stratix II devices. The logic element (LE) count is based on this number of ALUTs.

(2) Yields the minimum area possible without using DSP blocks.

Chapter 2. Getting Started

System Requirements

The instructions in this section require the following hardware and software:

- A PC running the Windows NT/2000/XP, Red Hat Linux 7.3 or 8.0, or Red Hat Enterprise Linux 3.0 operating system; or a Sun workstation running the Solaris 7 or 8 operating system
- The Quartus® II software version 4.1 or higher
- An Altera-supported VHDL or Verilog HDL simulator (optional).

Design Flow

To evaluate the Color Space Converter (CSC) MegaCore® function using the OpenCore® Plus feature, the design flow involves the following steps:

1. Obtain and install the CSC MegaCore function.
2. Create a custom variation of the CSC MegaCore function using IP Toolbench.

   IP Toolbench is a toolbar from which you can quickly and easily view documentation, specify parameters, and generate all of the files necessary for integrating the parameterized MegaCore function into your design. You can launch IP Toolbench from within the Quartus II software.

3. Implement the rest of your design using the design entry method of your choice.
4. Use the IP functional simulation model generated by IP Toolbench to verify the operation of your design.

For more information on IP functional simulation models, refer to the white paper entitled *Using IP Functional Simulation Models to Verify Your System Design*.

5. Use the Quartus II software to compile your design.

You can generate an OpenCore Plus time-limited programming file, which you can use to verify the operation of your design in hardware for a limited time. For more information on OpenCore Plus hardware evaluation using the CSC MegaCore function, see "OpenCore Plus Time-Out Behavior" on page 3–2, and *AN 320: OpenCore Plus Evaluation of Megafunctions*.

6. Purchase a license for the CSC MegaCore function.

Once you have purchased a license for the CSC MegaCore function, the design flow involves the following additional steps:

1. Set up licensing.
2. Generate a programming file for the Altera device(s) on your board.
3. Program the Altera device(s) with the completed design.
4. Perform design verification.

Obtain & Install the CSC MegaCore Function

To begin using the CSC MegaCore function you must obtain the CSC MegaCore function file, and install it on your computer. Altera® MegaCore functions can be installed from the MegaCore IP Library CD-ROM either during or after Quartus II installation, or downloaded individually from the Altera web site and installed separately.

The following instructions describe the process of downloading and installing the CSC MegaCore function. If you have already installed the CSC MegaCore function from the MegaCore IP Library CD-ROM, skip to "Directory Structure" on page 2–4.

Download the CSC MegaCore Function

If you have Internet access, you can download MegaCore functions from Altera's web site at www.altera.com. Follow the instructions below to obtain the CSC MegaCore function via the Internet. If you do not have Internet access, you can obtain the CSC MegaCore function from your local Altera representative.

1. Point your web browser to www.altera.com/ipmegastore.
2. Type `CSC` in the **IP MegaSearch** box.
3. Click **Go**.

4. Choose the Altera **Color Space Converter** from the search results page. The product description web page displays.

5. Click **Download Free Evaluation** on the top right of the product description web page.

6. Complete the registration form and click **Submit Request**.

7. Read the Altera MegaCore license agreement, turn on the **I have read the license agreement** check box, and click **Proceed to Download Page**.

8. Follow the instructions on the CSC MegaCore function download and installation page to download the MegaCore function and save it to your hard disk.

There is a specific MegaCore function download file for each supported operating system.

Install the CSC MegaCore Function Files

The following instructions describe how you install the CSC MegaCore function on computers running the Windows, Solaris, or Linux operating systems.

Windows

To install the CSC MegaCore function on a PC running the Windows operating system, follow these steps:

1. Choose **Run** (Start menu).

2. Type <*path*>`\csc-v2.2.0.exe` ↵

   where <*path*> is the location of the downloaded MegaCore function installation executable.

3. Click **OK**. The CSC MegaCore function installation dialog box appears. Follow the on-screen instructions to finish installation.

Solaris & Linux

Follow these steps to install the CSC MegaCore function on a computer running supported versions of the Solaris or Linux operating systems:

1. Move the compressed files to the desired installation directory and make that directory your current directory.

2. Decompress the package by typing the following command:

   `gunzip -d csc-v2.2.0_<operating system>.tar.gz` ↵

   where <*operating system*> is either `solaris` or `linux`.

3. Extract the package contents by typing the following command:

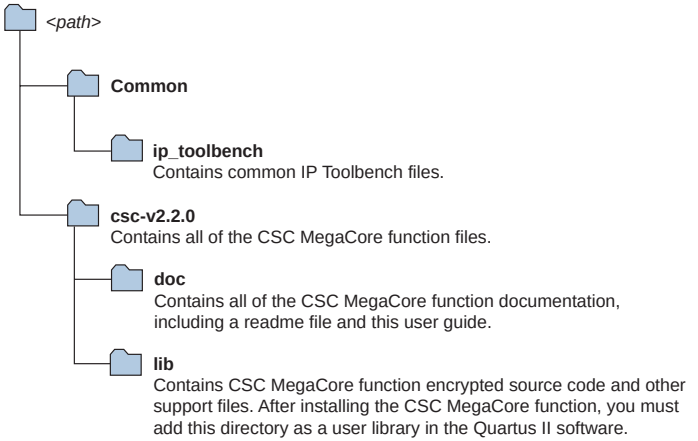
   `tar -xvf csc-v2.2.0_<operating system>.tar` ↵

   where <*operating system*> is either `solaris` or `linux`.

Directory Structure

Figure 2–1 shows the directory structure for the CSC MegaCore function, where <*path*> is the CSC MegaCore function installation directory.

Figure 2–1. CSC MegaCore Function Directory Structure


<path>
Common
ip_toolbench
Contains common IP Toolbench files.
csc-v2.2.0
Contains all of the CSC MegaCore function files.
doc
Contains all of the CSC MegaCore function documentation, including a readme file and this user guide.
lib
Contains CSC MegaCore function encrypted source code and other support files. After installing the CSC MegaCore function, you must add this directory as a user library in the Quartus II software.

CSC MegaCore Function Walkthrough

This walkthrough explains how to create a custom variation of a CSC MegaCore function using IP Toolbench and the Quartus II software on a PC running Windows. When you are finished generating a custom variation of a CSC MegaCore function, you can incorporate it into your overall project.

This walkthrough consists of these steps:

- "Create a New Quartus II Project" on page 2–5
- "Launch IP Toolbench" on page 2–6
- "Step 1: Parameterize" on page 2–8
- "Step 2: Set Up Simulation" on page 2–12
- "Step 3: Generate" on page 2–14

Create a New Quartus II Project

Before you begin, you must create a new Quartus II project. With the **Quartus II New Project Wizard**, you specify the working directory for the project, assign the project name, and designate the name of the top-level design entity. You will also specify the CSC MegaCore function user library. To create a new project, follow these steps:

1. Choose **Programs > Altera > Quartus II** <*version*> (Windows Start menu) to run the Quartus II software.
2. Choose **New Project Wizard** (File menu).
3. Click **Next** in the introduction (the introduction will not display if you turned it off previously).
4. Specify the working directory for your project. This walkthrough uses the directory **c:\csc_example**.
5. Specify the name of the project. This walkthrough uses **csc_example**.
6. Click **Next**.
7. Click **User Library Pathnames**.
8. Specify <*path*>**\csc-v2.2.0\lib** in the **Library name** box, where <*path*> is the directory in which you installed the CSC MegaCore function. The default installation directory is **c:\MegaCore**.
9. Click **Add**.
10. Click **OK**.
11. Click **Next**.
12. In the **Family** list, select Stratix. Under **Do you want to select a specific device?** select **No**.

13. Click **Finish**.

You have finished creating your new Quartus II project.

Launch IP Toolbench

To launch IP Toolbench in the Quartus II software, follow these steps:

1. Start the MegaWizard® Plug-In Manager by choosing **MegaWizard Plug-In Manager** (Tools menu). The **MegaWizard Plug-In Manager** dialog box is displayed.

   Refer to the Quartus II Help for more information on how to use the MegaWizard Plug-In Manager.

2. Specify that you want to create a new custom megafunction variation and click **Next**.

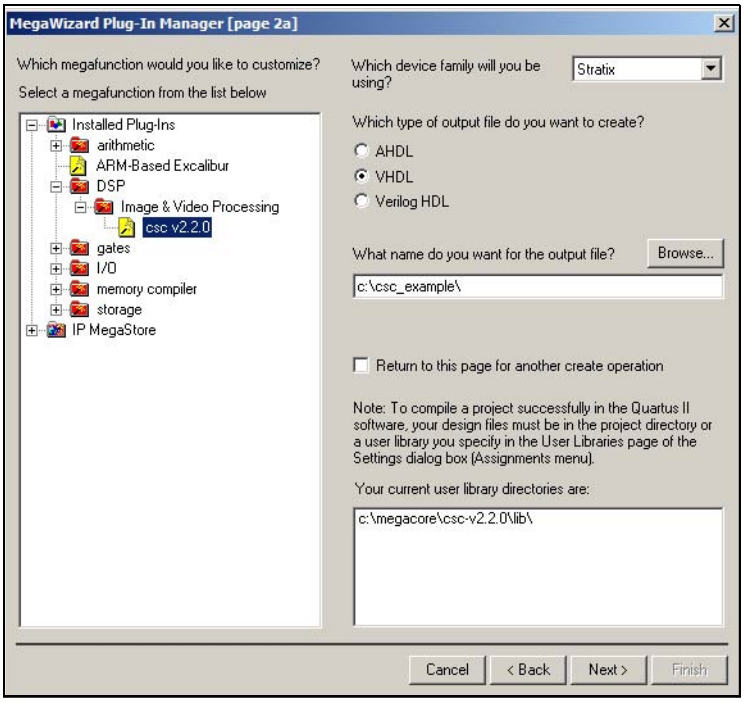
3. Expand the **DSP** folder under **Installed Plug-Ins** by clicking the **+** icon next to the name, then expand the **Image & Video Processing** folder the same way.

4. Select **csc-v2.2.0** under **Image & Video Processing**.

5. Choose the output file type for your design; the wizard supports AHDL, VHDL, and Verilog HDL. For this walkthrough choose **VHDL**.

6. Specify **c:\csc_example** for the directory name, and **csc_example** for the output file variation name. Figure 2–2 shows the wizard after you have made these settings.

Figure 2–2. MegaWizard Plug-In Manager Page 2a


MegaWizard Plug-In Manager [page 2a]
Which megafunction would you like to customize?
Select a megafunction from the list below
Installed Plug-Ins
arithmetic
ARM-Based Excalibur
DSP
Image & Video Processing
csc v2.2.0
gates
I/O
memory compiler
storage
IP MegaStore
Which device family will you be using?
Stratix
Which type of output file do you want to create?
AHDL
VHDL
Verilog HDL
What name do you want for the output file?
Browse...
c:\csc_example\
Return to this page for another create operation
Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory or a user library you specify in the User Libraries page of the Settings dialog box (Assignments menu).
Your current user library directories are:
c:\megacore\csc-v2.2.0\lib\
Cancel
< Back
Next >
Finish

7. Click **Next** to launch IP Toolbench for the CSC MegaCore function (Figure 2–3).

Figure 2–3. IP Toolbench

Step 1: Parameterize

To create a custom variation of a CSC MegaCore function, follow these steps:

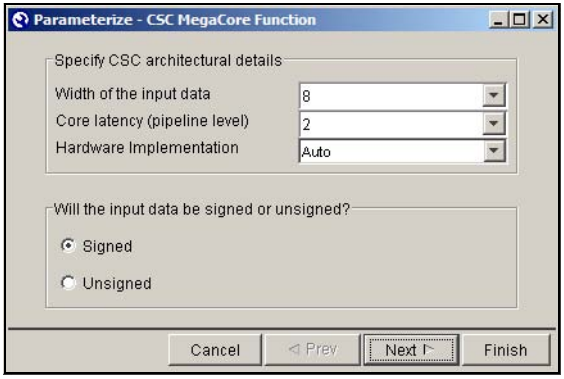
Refer to "Parameters" on page 3–3 for more information about the parameters set during this walkthrough.

1. Click **Step 1: Parameterize** in IP Toolbench. The first settings page of the **Parameterize - CSC MegaCore Function** wizard begins (Figure 2–4). This page allows you to specify:

   - The bit width of the input data bus
   - The core latency or pipeline level
   - The hardware implementation (e.g., *Multipliers using LUTs* or *Multipliers using DSP blocks*)
   - Whether the input bus is signed or unsigned (the signed representation uses the two's complement numbering scheme)

For this walkthrough, use the default settings (Figure 2–4).

Figure 2–4. First Settings Page of the Parameterize CSC MegaCore Function Wizard



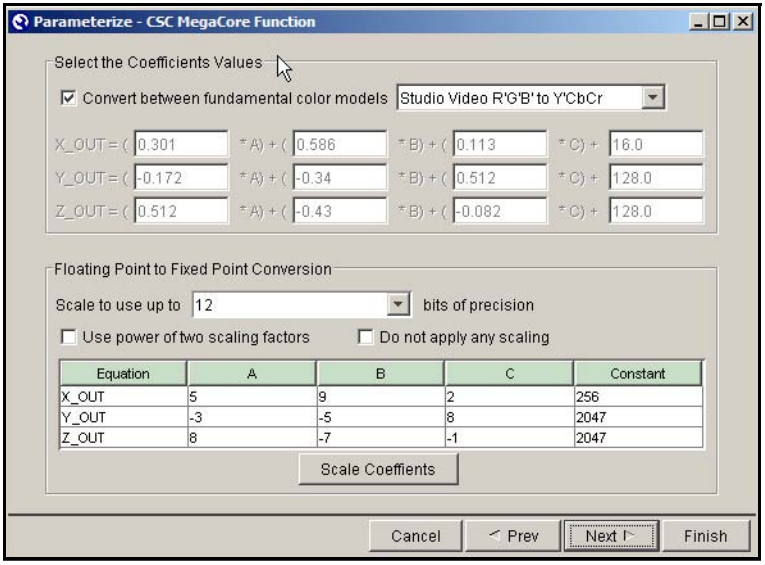
2. Click **Next**. The next page displays (Figure 2–5). This page allows you to specify the coefficients for the color space conversion matrix transform.

   Under **Select the Coefficient Values**, you can use a predefined function by checking **Convert between fundamental color models** and selecting from the preset color space conversions in the drop-down list. Or uncheck **Convert between fundamental color models** and manually enter or edit specific values to define your own custom transform matrix.

Under **Floating Point to Fixed Point Conversion**, you can optionally scale the transform functions. You can specify from 4 to 28 bits of precision, choose to use only power-of-two scaling factors, or no scaling at all. Click **Scale Coefficients** to see the resulting new values in the array.

For this walkthrough, use the default settings.

Figure 2–5. Second Settings Page


Parameterize - CSC MegaCore Function
Select the Coefficients Values
Convert between fundamental color models
Studio Video R'G'B' to Y'CbCr
X_OUT = (0.301 * A) + (0.586 * B) + (0.113 * C) + 16.0
Y_OUT = (-0.172 * A) + (-0.34 * B) + (0.512 * C) + 128.0
Z_OUT = (0.512 * A) + (-0.43 * B) + (-0.082 * C) + 128.0
Floating Point to Fixed Point Conversion
Scale to use up to 12 bits of precision
Use power of two scaling factors
Do not apply any scaling
Equation A B C Constant
X_OUT 5 9 2 256
Y_OUT -3 -5 8 2047
Z_OUT 8 -7 -1 2047
Scale Coeffients
Cancel
Prev
Next
Finish

3. Click **Next**. The third settings page displays (Figure 2–6). This page allows you to define the resolution of the X_OUT, Y_OUT, and Z_OUT signals generated by the CSC MegaCore function.

   You can specify whether to use **Full Precision** or **Limited Precision** for these output signals.

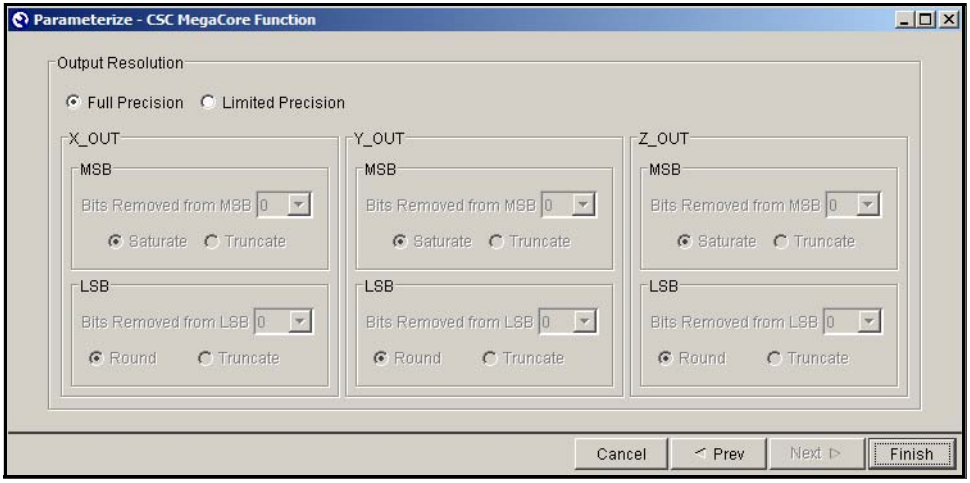
   The CSC MegaCore function determines the bit width of the output based on the bits of precision and the bit width of the input. These two parameters define a range of maximum positive and negative output values.

   The CSC MegaCore function extrapolates the number of bits required to represent that range of values. For **Full Precision**, you must use this number of bits in your system. If you choose **Limited Precision**, the wizard gives you the option of truncating or saturating the most significant bit (MSB) and/or rounding or truncating the least significant bit (LSB). Saturation and rounding are non-linear operations.

   For this walkthrough select **Full Precision.**

4. Click **Finish** to complete the parameterization of your CSC MegaCore function variation.

Figure 2–6. Third Settings Page



Step 2: Set Up Simulation

An IP functional simulation model is a cycle-accurate VHDL or Verilog HDL model file produced by the Quartus II software (version 3.0 or higher). It allows for fast functional simulation of IP using industry-standard VHDL and Verilog HDL simulators.

You may only use these simulation model output files for simulation purposes and expressly not for synthesis or any other purposes. Using these models for synthesis creates a non-functional design.

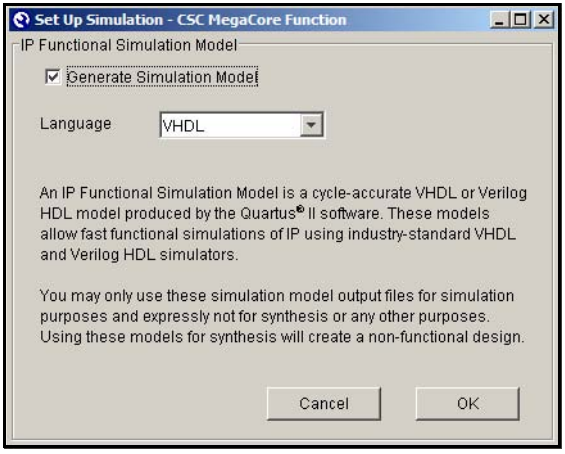
To generate an IP functional simulation model for your MegaCore function, follow these steps:

1. Click **Step 2: Set Up Simulation** in IP Toolbench (Figure 2–7).

Figure 2–7. Set Up Simulation

2. Turn on **Generate Simulation Model** (Figure 2–8).

Figure 2–8. Generate Simulation Model


Set Up Simulation - CSC MegaCore Function
IP Functional Simulation Model
Generate Simulation Model
Language
VHDL
An IP Functional Simulation Model is a cycle-accurate VHDL or Verilog HDL model produced by the Quartus® II software. These models allow fast functional simulations of IP using industry-standard VHDL and Verilog HDL simulators.
You may only use these simulation model output files for simulation purposes and expressly not for synthesis or any other purposes. Using these models for synthesis will create a non-functional design.
Cancel
OK

3. Choose **VHDL** in the **Language** list.

4. Click **OK**.

Step 3: Generate

To generate your MegaCore function, follow these steps:

1. Click **Step 3: Generate** in IP Toolbench (Figure 2–9).

Figure 2–9. IP Toolbench—Generate

2. The generation report lists the design files that IP Toolbench creates (Figure 2–10). Click **Exit**.

Figure 2–10. Generation

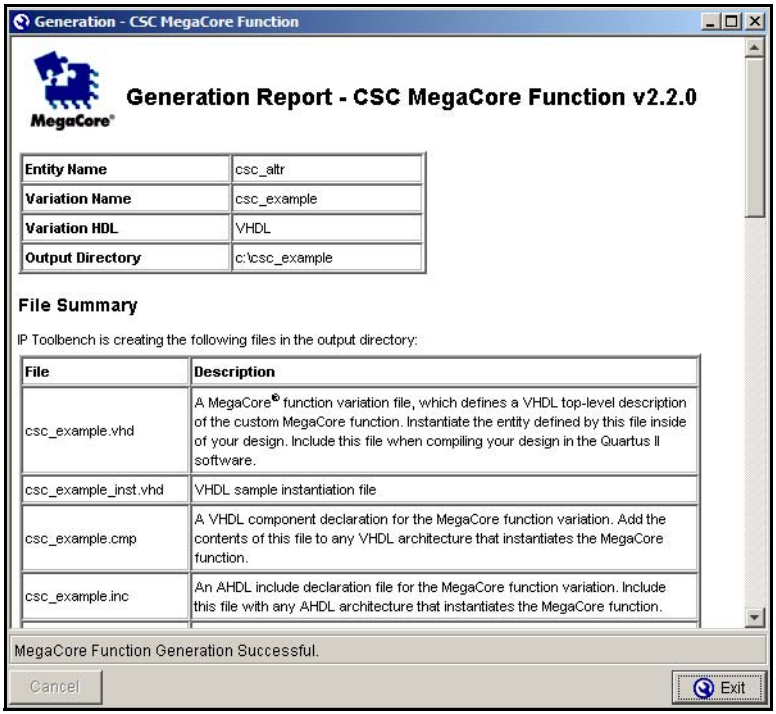

Generation - CSC MegaCore Function
MegaCore
Generation Report - CSC MegaCore Function v2.2.0
Entity Name
csc_altr
Variation Name
csc_example
Variation HDL
VHDL
Output Directory
c:\csc_example
File Summary
IP Toolbench is creating the following files in the output directory:
File
Description
csc_example.vhd
A MegaCore® function variation file, which defines a VHDL top-level description of the custom MegaCore function. Instantiate the entity defined by this file inside of your design. Include this file when compiling your design in the Quartus II software.
csc_example_inst.vhd
VHDL sample instantiation file
csc_example.cmp
A VHDL component declaration for the MegaCore function variation. Add the contents of this file to any VHDL architecture that instantiates the MegaCore function.
csc_example.inc
An AHDL include declaration file for the MegaCore function variation. Include this file with any AHDL architecture that instantiates the MegaCore function.
MegaCore Function Generation Successful.
Cancel
Exit

Table 2–1 describes the IP Toolbench-generated files.

Table 2–1. IP Toolbench-Generated Files

Extension	Description
.vhd, or **.v**	A MegaCore function variation file that defines a VHDL or Verilog HDL top-level description of the custom MegaCore function. Instantiate the entity defined by this file inside of your design. Include this file when compiling your design in the Quartus II software.
.cmp	A VHDL component declaration file for the MegaCore function variation. Add the contents of this file to any VHDL architecture that instantiates the MegaCore function.
.inc	An AHDL include declaration file for the MegaCore function variation. Include this file with any AHDL architecture that instantiates the MegaCore function.
_bb.v	Verilog HDL black-box file for the MegaCore function variation. Use this file when using a third-party EDA tool to synthesize your design.
.bsf	A Quartus II symbol file for the MegaCore function variation. Use this file in the Quartus II block diagram editor.
.html	A MegaCore function report file.
.vo or **.vho**	A VHDL or Verilog HDL IP functional simulation model.
_inst.vhd or **_inst.v**	A VHDL or Verilog HDL sample instantiation file.

You can now integrate your CSC MegaCore function variation into your design and simulate and compile.

Simulate the Design

You can simulate your design using the IP Toolbench-generated VHDL and Verilog HDL IP functional simulation models. The IP functional simulation model is the VO or VHO file you specified in "Step 2: Set Up Simulation" on page 2–12. Compile the file in your simulation environment and perform functional simulation of your custom CSC MegaCore function.

For more information on IP functional simulation models, refer to the white paper entitled *Using IP Functional Simulation Models to Verify Your System Design*.

Compile the Design

You can use the Quartus II software to compile your design. Refer to Quartus II Help for instructions on performing compilation.

Refer to Quartus II Help (**F1**) or the *Introduction to Quartus II Handbook* for further instructions on compiling and analyzing your design.

Program a Device

After you have compiled your design, program your targeted Altera device, and verify your design in hardware.

With Altera's free OpenCore Plus evaluation feature, you can evaluate the CSC MegaCore function before you purchase a license. OpenCore Plus evaluation allows you to generate an IP functional simulation model, and produce a time-limited programming file.

For more information on IP functional simulation models, refer to the white paper entitled *Using IP Functional Simulation Models to Verify Your System Design*.

For more information on OpenCore Plus hardware evaluation using the CSC MegaCore function, see "OpenCore Plus Time-Out Behavior" on page 3–1 and *AN 320: OpenCore Plus Evaluation of Megafunctions*.

Set Up Licensing

You need to purchase a license for a CSC MegaCore function only when you are completely satisfied with its functionality and performance, and want to take your design to production.

After you purchase a license for a CSC MegaCore function, you can request a license file from the Altera web site at www.altera.com/licensing and install it on your computer. When you request a license file, Altera e-mails you a **license.dat** file. If you do not have Internet access, contact your local Altera representative.

To install your license, you can either append the license to your existing Quartus II license file or you can specify the CSC MegaCore function's **license.dat** file in the Quartus II software.

Before you set up licensing for a CSC MegaCore function, you must already have the Quartus II software installed and liscensed on your computer.

Append the License to Your license.dat File

To append the license, follow these steps:

1. Close the following software if it is running on your PC:
   - Quartus II software
   - MAX+PLUS® II software
   - LeonardoSpectrum™ synthesis tool
   - Synplify software
   - ModelSim® simulator

2. Open the CSC MegaCore function license file in a text editor. The file should contain one FEATURE line, spanning 2 lines.

3. Open your Quartus II **license.dat** file in a text editor.

4. Copy the FEATURE line from the CSC MegaCore function license file and paste it into the Quartus II license file.

   Do not delete any FEATURE lines from the Quartus II license file.

5. Save the Quartus II license file.

   When using editors such as Microsoft Word or Notepad, ensure that the file does not have extra extensions appended to it after you save (e.g., **license.dat.txt** or **license.dat.doc**). Verify the filename in a DOS box or at a command prompt.

Specify the License File in the Quartus II Software

To specify the CSC MegaCore function's license file, follow these steps:

1. Altera recommends that you give the file a unique name, e.g., **altera-csc_license.dat**.

2. Start the Quartus II software.

3. Choose **License Setup** (Tools menu). The **Options** dialog box opens to the **License Setup** page.

4. In the **License file** box, add a semicolon to the end of the existing license path and filename.

5. Type the path and filename of the CSC MegaCore function license file after the semicolon.

   Do not include any spaces either around the semicolon or in the path/filename.

6. Click **OK** to save your changes.

Chapter 3. Specifications

Functional Description

A three-dimensional color space is defined as a mathematical representation of a set of colors, where each color is mapped to three coordinates. The Color Space Converter (CSC) MegaCore® function transforms a color from one three-dimensional color space to another by multiplying the tri-stimulus value by a 3 x 4-matrix transform.

The CSC MegaCore function uses this equation to convert data from one color space to another:

$$\begin{bmatrix} X\_OUT \\ Y\_OUT \\ Z\_OUT \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ 1 \end{bmatrix}$$

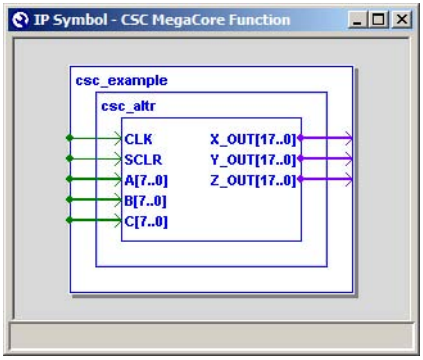
Because the inputs are multiplied by constant values, the look-up table (LUT) architecture of Altera® SRAM-based FPGAs is ideal for implementing the conversion equations. Pre-computing partial products and storing them in look-up tables can provide a smaller, faster implementation than one that can be realized with soft multipliers.

See "References" on page 3–6 for more information on color spaces and converting between them.

Signals

Figure 3–1 shows the symbol and signals for the CSC MegaCore Function produced by clicking **Display Symbol** in IP Toolbench.

Figure 3–1. CSC MegaCore function Symbol



OpenCore Plus Time-Out Behavior

OpenCore® Plus hardware evaluation can support the following two modes of operation:

- *Untethered*—the design runs for a limited time
- *Tethered*—requires a connection between your board and the host computer. If tethered mode is supported by all megafunctions in a design, the device can operate for a longer time or indefinitely

All megafunctions in a device time out simultaneously when the most restrictive evaluation time is reached. If there is more than one megafunction in a design, a specific megafunction's time-out behavior may be masked by the time-out behavior of the other megafunctions.

For the CSC and other MegaCore functions, the untethered timeout is one hour, and the tethered timeout value is indefinite.

The output signals `X_OUT`, `Y_OUT`, and `Z_OUT` go low when the evaluation time for the CSC MegaCore function expires.

For more information on OpenCore Plus hardware evaluation, see "OpenCore Plus Evaluation" on page 1–3 and *AN 320: OpenCore Plus Evaluation of Megafunctions*.

Parameters

You configure the CSC MegaCore function using the **Parameterize - CSC MegaCore Function** wizard interface (see "CSC MegaCore Function Walkthrough" on page 2–4). Tables 3–1, 3–2, and 3–3 list the parameters relevant to each page in the wizard.

Table 3–1. Parameters on the First Page of the Parameterize - CSC MegaCore Function Wizard

Parameter	Value (default)	Description
Width of the input data	2 - 32 (8)	Defines the width of the input bus carrying the color data.
Core latency (pipeline level)	0 - 9 (2)	Affects the time required to process the color data. A higher number yields a faster but larger architecture.
Hardware Implementation	Auto (default) Distributed Arithmetic in LUTs Multipliers using LUTs Multipliers using DSP blocks	Determines the hardware implementation used for the CSC. May be the default implementation for the targeted device family, or a specific style as specified. For details, refer to *AN 306: Techniques for Implementing Multipliers in Stratix®, Stratix GX & Cyclone™ Devices*.
Input data signed or unsigned?	Signed (default) Unsigned	Identifies whether the input data is signed or unsigned.

Table 3–2. Parameters on the Second Page of the Parameterize - CSC MegaCore Function Wizard

Parameter	Value (default)	Description
Convert between fundamental color models	on or off	Turn on to use a set of coefficients based on the preset color model conversion you then choose from the drop-down list: - Studio Video RGB to YCbCr - YCbCr to Studio Video RGB - Computer RGB to YCbCr - YIQ to YUV - YUV to Computer RGB - Computer RGB to YUV Turn off to edit the floating-point value of each coefficient of the color space conversion equations. See "References" on page 3–6 for more information on color model conversion.
Scale to use up to X bits of precision	4 - 28 (12)	Allows you to convert the floatingpoint coefficients defined above to fixed-point coefficients using the number of bits of precision that you specify. This conversion consists of multiplying all the floating-point coefficients with a common scaling factor and casting the floating-point multiplication result to a two's complement integer (a fixed-point coefficient). The scaling factor is defined so that the floating-point dynamic range (min-max value) gets mapped to the coefficient bit-width dynamic range (min-max).
Use only power of two scaling factors	on or off	Turn on to round the scaling factor to the nearest power-of-two integer value
Do not apply any scaling	on or off	Turn on to set the scaling factor to 1.0

Table 3–3. Parameters on the Third Page of the Parameterize - CSC MegaCore Function Wizard		
Parameter	**Value**	**Description**
Output Resolution	Full Precision or Limited Precision	Choose **Full Precision** to use the full range of the output bits without saturation, truncation, or roundoff. Choose **Limited Precision** to constrain the output bits.
X_OUT, Y_OUT, Z_OUT: MSB and LSB	Bits removed, Saturate or Truncate	To constrain the output bits, for each output specify the number of bits to remove through saturation or truncation for its most-significant-bit or least-significant-bit.

Signals

Table 3–4 describes the external signals of the CSC MegaCore function. (Refer to the CSC symbol diagram, Figure 3–1 on page 3–2.)

Table 3–4. CSC Signals		
Signal	**Direction**	**Description**
CLK	Input	The system clock.
SCLR	Input	The synchronous clear signal, which is active at 1.
A[] B[] C[]	Input	The input buses.
X_OUT[] Y_OUT[] Z_OUT[]	Output	The converted data.

MegaCore Verification

Before releasing a version of the CSC MegaCore function, Altera runs comprehensive regression tests to verify its quality and correctness.

Custom variations of the CSC MegaCore function are generated to exercise its various parameter options, and the resulting simulation models are thoroughly simulated and the results verified against bit-accurate master simulation models.

References

Altera application notes, white papers, and user guides providing more detailed explanations of how to effectively design with MegaCore functions and the Quartus® II software are available at the Altera web site (www.altera.com).

The following third-party references provide technical information on color spaces and the transformations between them.

- C. Shi, and R. W. Brodersen, *Floating-point to fixed-point conversion with decision errors due to quantization*, Proceeding, IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2004, Montreal, Canada
- Changchun Shi and Robert W. Brodersen, *An Automated Floating-point to Fixed-point Conversion Methodology*, presented at ICASSP 2003
- Arthur B. Williams and Fred J. Taylor, *Electronic Filter Design Handbook*, McGraw Hill, 3rd edition (October 1995)
- Jack, Keith, *Video Demystified, A Handbook for the Digital Engineer*, Second Edition. Solana Beach: Hightext Publications, 1996